A Novel Protocol to Allow Revocation of Votes in a Hybrid Voting System

Oliver Spycher
University of Fribourg
Department of Informatics
CH-1700 Fribourg, Switzerland
E-mail: oliver.spycher@unifr.ch

Rolf Haenni
Bern University of Applied Sciences
Engineering and Information Technology
CH-2501 Biel, Switzerland
E-mail: rolf.haenni@bfh.ch

***Abstract*—A hybrid voting system allows voters to revoke their electronic vote at the polling station. This approach is meant to provide full individual and universal verifiability without introducing the threats of vote buying or voter coercion. Such an integration of traditional and electronic voting systems requires the voters' ability to prove whether they have already voted electronically, and if so, to show which of all the electronic votes published on the public bulletin board is theirs.**

This paper proposes in full cryptographic detail a novel e-voting protocol that allows voters to unambiguously show and prove to voting officials at the polling station if they have cast an electronic vote. If this is the case, the voters can use their secret credentials to locate their votes on the public bulletin board without giving up the secrecy of the credentials. Remarkably, our protocol enables them to do so, even if their votes have been cast by a third party that got hold of their credentials. We thus address the hardest possible attack on a voter's right to vote. Furthermore, unlike pure e-voting systems, our protocol allows the hybrid system to provide coercion-resistance even when voters are allowed to vote for write-in candidates.

Our approach is meant to appeal to governments that aim at offering voters the choice between two channels for casting votes, rather than fully replacing their traditional paper-based voting scheme with an e-voting system.[1]

I. Introduction

In consideration of the complexity and manifold vulnerabilities of today's computers and networks, most governments pursue a cautious strategy in introducing electronic means into processes that are so fundamental for running their democracy. Their reservation is particularly distinctive if the technology involves components that are not under their control. The number of countries experimenting with electronic voting over the Internet is therefore still marginal. Estonia and Switzerland, two of the few pioneering countries in Internet elections and referendums (we shall use the general term voting), follow the strategy to slowly increase the number of electronic votes over the years [1]. The idea behind keeping this shift at a slow pace is to limit the risk and consequences of fraud in the early stage of the respective project. The legitimacy of such concerns has been demonstrated by the negative e-voting experience in the Netherlands, where all nationwide e-voting activities have been stopped in 2007, after the vulnerability of the deployed system had been exposed in public [2]. In the foreseeable future, traditional and electronic voting systems are therefore expected to live side by side for quite some time. In this paper we introduce a protocol which is applicable to that very setting.

Offering the voting community two channels for casting their votes requires some care. Clearly, a mechanism is required to prevent voters from casting multiple votes, particularly one vote per channel. A voting system that complies with this minimal requirement we call *integrated voting system*. To find acceptance among voters, a new integrated voting system should at least level with the security standards of the existing voting system. Note, that the security level of an integrated voting system is directly determined by the weaker of its subsystems.

In Section II, we show the requirements on e-voting systems as they are commonly postulated. We explain why it is so difficult to offer *coercion-resistance* with pure e-voting systems [3]. Since an electronic sub-system vulnerable to coercion would clearly undermine the security level of the overall integrated system, we proposed in [4] the concept of a *hybrid voting system*, in which voters are allowed to revoke their electronic votes at the polling station. For this, we require the electronic component of a hybrid voting system to hold properties that differ from the ones of a pure e-voting system. The notion of hybrid voting systems thus opens new possibilities for protocol designers. Remarkably, coercion-resistance is an inherent consequence of the definition of hybrid systems.

In Section III and Section IV, we present a novel e-voting protocol that can be used as the electronic sub-system of a hybrid system. It is based on an anonymous authentication mechanism and guarantees that the voters are always able to unambiguously locate their votes on the public bulletin board during the revocation procedure. An overview of the protocol is given in Section III, and the corresponding formal details are provided in Section IV. In Section V, we show that the protocol satisfies the corresponding requirements as described in Section II, and Section VI concludes the paper.

[1]Research supported by the *Hasler Foundation* (project No. 09037) and the *Mittelbauförderung* of the Bern University of Applied Sciences.

II. Hybrid Voting Systems

A functioning democracy depends on its citizens' trust in their political voting system. This applies to remote electronic voting systems which allow voters to cast their votes through the Internet, just as much as to their traditional paper-based counterparts. In both cases, citizens must be convinced that the system is sufficiently secure and invulnerable. Subsection II-A introduces and explains the requirements on e-voting systems that are often postulated. They are directly derived from traditional paper-based voting systems that are usually considered to be sufficiently secure and trustworthy. In Subsection II-B, we discuss the difficulty of offering coercion-resistance with pure e-voting systems, and in Subsection II-C, we show how to overcome that problem with hybrid voting systems.

A. Requirements on E-Voting Systems

For an e-voting system to be secure, it has to be implemented according to an intrinsically secure design. Despite the complexity of designing and implementing such a system, some criteria seem to be unanimously accepted as the core security requirements for e-voting systems [5], [6].

- *Accuracy:* A system is accurate if votes cast can not be altered (integrity), valid votes can not be eliminated from the final tally (completeness), and invalid votes are not counted in the final tally (soundness).
- *Democracy:* A system is democratic if only authorized voters can vote (eligibility) and authorized voters can only vote once (uniqueness).
- *Privacy:* A system is private if no vote cast can be linked to its voter, neither by voting authorities nor anyone else (anonymity), and no voter can prove that he or she voted in a particular way (receipt-freeness).
- *Verifiability:* A system is *individually verifiable* if voters can independently verify that their own votes have been counted correctly in the final tally. A system is *universally verifiable*, if voters can independently verify that all votes cast have been counted correctly in the final tally.
- *Fairness:* A system is fair if no intermediate results can be obtained before the voting period ends.
- *Coercion-Resistance:* A system is coercion-resistant if it is immune against *vote buying* and *voter coercion*.

The terms *coercion-resistance*, *vote buying* and *voter coercion*, as well as the reasons why they have caused severe headaches among protocol designers are further elaborated in the next subsection.

B. Coercion-Resistance versus Individual Verifiability

Whether or not a system has actually implemented required security features is not necessarily evident to the voters. If they feel that their votes may not even reach the final tally, they might fully restrain from voting electronically and tend to cast their votes in the traditional way, a means of casting votes still likely to be available in the near future. By doing so, they witness the vote reaching the body of the possibly transparent ballot-box. Some countries even allow voters to attend the tallying procedure and thus to witness the consideration of their votes in the final outcome. To establish a similar level of voters' trust in e-voting systems, it is imperative to give them access to some information that confirms the correct casting of their votes in a convincing way. This confirmation is meant to provide aforementioned *individual verifiability*, a precondition to trustworthiness of voting systems. The existence of such a confirmation may thus seem like a feature, but since it will generally also convince any third party that a particular vote was cast, it disallows voters to deceive others about their votes. Such information is thus called a voter's *receipt* [7]. Its existence is a violation of the voter's privacy, because it opens doors to the following two types of fraud, in which the adversary gets the voter to vote in a prescribed way [8].

- *Vote Buying:* The voter will be rewarded by the *vote buyer* for voting in a particular manner. To receive the reward, the voter might actively co-operate with the vote buyer, e.g. by deviating from the normal voting procedure to construct a receipt.
- *Voter Coercion:* The voter is threatened by a *coercer* to vote in a particular manner. Here, the voter will only consent to co-operate with the coercer as long as the threat is perceived as real.

Note, that both forms of exploiting a voting system are largely scalable in an electronic environment. A vote buyer could simply set up a web site explaining the conditions for making easy money, while a coercer could easily post his threats to thousands of voters. In both cases, the attack is only interesting to potential adversaries as long as voters are able to prove them how they voted. Without a receipt, a corrupted voter could simply lie about the vote cast, i.e., the motivation of an adversary even launching such an attack in the first place is likely to be as low as with paper-based votes.

Clearly, it must be a primary objective to establish an e-voting system that is immune to all sorts of vote buying and voter coercion attacks, including those in which the adversary gets the voter to abstain from voting or to vote at random. Systems blessed with that immunity are called *coercion-resistant* [3], [9]. Note that coercion-resistance is stronger than mere *receipt-freeness* [7], [10], which alone does not prevent adversaries from getting voters to abstain from voting. In the literature, there are many suggestions for receipt-free or coercion-resistant systems, but most of them rely on unrealistic technical assumptions [7], [9], [11]–[17].

C. Hybrid Voting Systems

Apart from receipt-freeness and thus coercion-resistance, the core security requirements as listed in Subsection II-A are addressed to a satisfactory degree by various known e-voting protocols. However, banning receipts from their systems to allow coercion-resistance, without compromising the inevitable *individual verifiability*, poses a great challenge and may even seem to be inherently infeasible at first sight. Yet, [3] proposes a system to solve just that problem. However, the scheme raises a few new technical questions that remain unanswered. Further, the scheme only grants for coercion-resistance in the case of referendums or elections with only few candidates to

choose from. If many candidates are eligible or if even *write-ins* are permitted, the decrypted vote will serve the voter as a receipt after tallying and thus subjects him to voter coercion and vote buying.

Hybrid voting systems on the contrary are designed to offer coercion-resistance in every thinkable mode of voting [4]. This is achieved by allowing voters to revoke their votes in the protecting environment of a polling station and replace it with a new, independent vote using the traditional paper-based voting channel. This approach enables individual voters to express their actual, unbiased political opinion.

If adversaries must assume that corrupted voters will usually revoke their votes, a hybrid system clearly provides coercion-resistance: an attack would simply seem too expensive. We believe that it is possible for governments to invoke that perception among adversaries, for instance by explicitly allowing voters to co-operate with vote buyers and coercers, however only as long as they revoke their biased vote.

In order for an e-voting protocol to define the electronic channel of a hybrid system, it needs to comply with the following requirements:

- *Proof of Eligibility:* Registered voters that abstained from casting an electronic vote need to be able to unambiguously prove to the voting officials that they are still eligible for casting their vote.[2] Assuming that the electronic voting phase ends before the traditional polling stations open, the minimal requirement for an integrated system is thus fulfilled.
- *Proof of Vote Ownership:* If their vote has been cast, voters need to be able to prove ownership of their (encrypted) electronic vote in the electronic ballot-box.

In the following sections, we present a protocol that satisfies these requirements. To satisfy *Proof of Vote Ownership*, the protocol guarantees that voters own respective receipts for the votes they own.[3] Notably, by requiring instead of banning receipts, we sharply depart from the mainstream approach of taking additional measures to make electronic voting systems receipt-free as a precondition to coercion-resistance.[4] We rather argue that the guaranteed existence of a receipt within the e-voting system allows coercion-resistance of the embedding hybrid system.

[2]By a *voter's vote* we consistently refer to the vote that was cast using his credentials. We say the credentials belong to the *owner of the vote*. We hereby address the situation where an adversary, for instance a vote buyer, gets hold of a voter's credentials.

[3]Two different revocation procedures are given in [4]. Since the protocol that we present here guarantees a receipt to vote owners, it allows the application of both procedures. By only requiring a *vote identifier* at revocation time, revocation is restricted to Procedure 1. Remarkably, small changes to the protocol introduce new features while still guaranteeing a *vote identifier* to vote owners. Particularly, a derivative of that protocol could offer a re-voting feature (last vote counts), which corresponds with the voting tradition of the Nordic countries.

[4]Due to the difficulty of excluding receipts from e-voting systems, many proposed systems simply accept the presence of receipts, hence their exposure to coercion and vote buying attacks. However, they generally only guarantee the ownership of a receipt to the party who cast the vote, not the owner of the vote. Thus, these systems will not meet this requirement.

III. Protocol Overview

In this section, we give a first overview of the proposed protocol. In Subsection III-A, we briefly discuss the cryptographic building blocks that are put to use, and in Subsection III-B, we give a high-level, informal description of the full protocol. This introductory subsection should motivate the thorough definitions of the protocol in Section IV and thus facilitate the reader's first approach.

A. Cryptographic Building Blocks

The protocol assumes several modern cryptographic building blocks. Apart from standard ElGamal encryption/decryption, we also need threshold cryptosystems, non-interactive zero-knowledge proofs of knowledge, anonymous authentication, mix networks, and anonymous channels. Some of these building blocks will be briefly described below.

ElGamal Cryptosystem: The ElGamal cryptosystem is based on a multiplicative cyclic group $(G, \cdot)$ of finite order q, for which the computational and the decisional Diffie-Hellman assumptions are believed to hold. The most common choice for such a group is a subgroup $G_q \subseteq \mathbb{Z}_p^*$ of order $q = (p-1)/k$, where p and q are large primes. The public parameters of an ElGamal cryptosystem are then p, q, and a generator g of G_q. An ElGamal key pair is a tuple (d, e), where the $d \in_R \mathbb{Z}_q$ is the randomly chosen private key and $e = g^d \in G_q$ the corresponding public key. If $M \in G$ denotes the message to encrypt, then the pair $(x, y) = (g^k, M \cdot e^k)$ is the encryption of M with randomness $k \in \mathbb{Z}_q$. For a given ElGamal encryption (x, y), M can be recovered by computing $M = \frac{y}{x^d}$.

Threshold Cryptosystems: A cryptosystem such as ElGamal is called threshold (t, n)-cryptosystem, if the private key to decrypt the message is shared among n parties, and if the number of parties required to cooperate in the decryption protocol exceeds a certain threshold $t < n$. A threshold version of the ElGamal cryptosystem results from sharing the private key with Shamir's secret sharing scheme. To avoid a trusted third party to generate the private key shares, it is possible to let the n parties execute a distributed key generation protocol [18].

Zero-Knowledge Proofs of Knowledge: A zero-knowledge proof allows a party to demonstrate to another party that a mathematical statement is true, but without revealing anything other than the truth of the statement itself. A particular class of zero-knowledge proofs are so-called *proofs of knowledge*, in which the prover demonstrates knowledge of the preimage ω of a public value $x = \phi(\omega)$, where $\phi : \mathcal{G} \to \mathcal{H}$ is a candidate one-way function. Such proofs can be constructed as non-interactive Σ-*protocols*, if ϕ is a homomorphism with a finite domain [19]. Two of the simplest and most frequently used instances of such Σ-protocols are the proof of knowledge of a discrete logarithm $\omega = \log_g x$ in a multiplicative group G of finite order q with generator g [20], or similarly, the proof of equality of two discrete logarithms $\omega = \log_g x = \log_h y$, where h is another generator of G [21]. In this paper, we use mere ZKP-notation to express that a non-interactive Σ-protocol is used as a means of proving. Moreover, ZKP-

notation expresses what is being proved. To prove the equality of two discrete logarithms, for instance as mentioned above, we would write $ZKP\big[(\omega) \ : \ (x = g^{\omega}) \ \wedge \ (y = h^{\omega})\big]$. More generally, Σ-protocols can be used to prove knowledge of preimages in zero-knowledge, that satisfies any composition of equations as in the example above, i.e., connected by logical $\wedge$ or $\vee$. Since ZKP-notation is understood intuitively, we do not give a formal definition here.

Anonymous Channels: An anonymous channel hides the correspondence between senders and their messages, i.e., the senders of the messages remain anonymous or untraceable. The most common realization of anonymous channels is based on *mix nets* [22]. A mix net consists of a sequence of servers, each of which receives a batch of input messages and produces a batch of output messages in a permuted (mixed) order.

Public Bulletin Board: A public bulletin board is a broadcast channel with memory. This means that everybody is allowed to append new entries and to read its content, but nobody is allowed to delete or to modify existing entries. Such a bulletin board may have the additional functionality of filtering out invalid or double entries, for example by checking the validity of an attached digital signature or proof of knowledge. The bulletin boards can be replicated in order to prevent them from being potential single points of failure.

B. Protocol Overview

- *Generation of Public and Secret Credentials:* As a precondition to a voting process, the protocol assumes the existence of a publicly readable voter roll. It can be thought of as a list that identifies all eligible voters. Each voter is assigned a *public credential* and the matching *secret credential*. The credential is kept secret by the voter. These values can be reused across multiple voting processes. A voter's public credential is associated with his entry in the voter roll and published. Without disclosing it, voters can prove that they own the secret credential that matches their public credential using a non-interactive Σ-protocol. On the other hand, it is computationally infeasible to calculate the secret credential that matches a voter's public credential.
- *Generation of Pseudonyms:* Given the list of public credentials as input, a publicly readable list of shuffled pseudonyms is generated before every voting event. Similarly as with public credentials, voters can prove that they own the secret credentials that match their pseudonyms. On the other hand, it is computationally infeasible to calculate the secret credential that matches a voter's pseudonym. Associating public credentials with their corresponding pseudonym is computationally only feasible when knowing the corresponding secret credential.
- *Vote Casting:* Voters use their credential to compute their pseudonym, the encryption of their vote, and a zero-knowledge proof that they have done so correctly, i.e., using the credential that matches the pseudonym in both computations. Clearly, only voters who know the credential that matches a pseudonym can do so. The pseudonym, the encrypted vote, and the proof are posted to the public bulletin board through an anonymous channel. If the proof holds against the sent values and if the supplied pseudonym is an element of the shuffled list, the vote and the proof are published on the board. By associating their vote with their pseudonyms, which is only possible when knowing the corresponding secret credential, voters authenticate themselves as eligible voters without disclosing their identity.[5]
- *Proofs of Eligibility and Ownership:* As described in Subsection II-C, the protocol must enable voters to prove that they have not cast an electronic vote. If they have cast an electronic vote, they must be able to identify the vote they have cast and prove their ownership. Both requirements are satisfied by the knowledge of their secret credential. At the polling station, voters authenticate themselves and identify their public credential on the public bulletin board. Further, they reveal the pseudonym that corresponds with their public credential and present a zero-knowledge proof to show that they have presented the correct pseudonym.[6] They can only do so using their secret credential. If there is no vote associated with that pseudonym, voters have proven their eligibility to cast their vote using the traditional paper-based infrastructure without prior revocation. If there is a vote associated with the pseudonym, the voter has proven ownership of that vote, i.e., to cast another vote using the paper-based infrastructure, it must first be revoked by following a revocation procedure described in [4].

IV. Detailed Protocol Definition

We divide the protocol into eight different steps, of which the first two need not to be repeated for every voting event.

A. Setup

The protocol involves four groups of players, each of which is responsible for designated tasks as described in the following subsections.

1) The group of eligible *voters* $\mathcal{V} = \{V_1, \ldots, V_m\}$, none of which are are assumed to be trustworthy.
2) The group of *registrars* $\mathcal{R} = \{R_1, \ldots, R_{n_r}\}$, of which at least $t_r \leq n_r$ are assumed to be trustworthy.
3) The group of *pseudonym producers* $\mathcal{P} = \{P_1, \ldots, P_{n_p}\}$, of which at least $t_p \leq n_p$ are assumed to be trustworthy.
4) The group of *talliers* $\mathcal{T} = \{T_1, \ldots, T_{n_t}\}$, of which at least $t_t \leq n_t$ are assumed to be trustworthy.

We collectively refer to registrars, pseudonym producers, and talliers by the term *voting authorities*.[7] Any intersection

[5] In the literature, this concept is sometimes called *anonymous authentication* [23], [24].

[6] Simply revealing the credential would compensate for the zero-knowledge proof. However, in that case voters would need to be assigned a new pair of public and secret credentials to meet the privacy requirement in subsequent voting events.

[7] For the sake of simplicity, we assume the members of voting authorities to be individuals. In reality, each group member could be an independent organization.

of groups can be non-void. Particularly, voters can work as registrars, pseudonym producers, or talliers at the same time. To simplify the formal notation, we assume that the size of each voting authority group is equal to n and that at least $t \leq n$ of their members are trustworthy ($n = n_r = n_p = n_t$ and $t = t_r = t_p = t_t$). Additionally, we assume that the threshold t of trustworthy authorities is strictly greater than $\frac{n}{2}$ (this is the best achievable threshold for a solution that provides both secrecy and robustness). Finally, we assume the presence of adversaries without explicitly formalizing them as a group.

Suppose that the voting authorities have agreed on a generator g of a subgroup $G_q \subseteq \mathbb{Z}_p^*$ of order q, such that p and $q = (p-1)/k$ are large primes (so-called *safe primes*). These values are used across multiple voting events.

We further assume the existence of a voter roll, an initially empty public bulletin board as a public communication channel, and an anonymous channel for casting the votes.

B. Generation of Public and Secret Credentials:

Objective: V_i knows his secret credential s_i and the corresponding public credential $S_i = g^{s_i}$ is published on the public bulletin board. These values can be reused across multiple voting events. Eligible voters are thus registered for e-voting.

Definition: For each is eligible voter $V_i \in \mathcal{V}$, the registrars $\mathcal{R}$ jointly create V_i's public credential $S_i = g^{s_i}$ and publish it on the public bulletin board associated with his entry in the voter roll. This can be done using a distributed key generation protocol as proposed in [18], where the knowledge of s_i is shared among the members of $\mathcal{R}$ such that at least t shares are required to compute s_i. The members of $\mathcal{R}$ pass their shares of s_i to V_i through a sufficiently secure channel (we assume that at least t members will do so). This could for example be done through the postal system or by V_i showing up at the registration offices for in-person authentication. The received shares allow V_i to efficiently compute s_i.

C. Generation of Pseudonyms

Objective: For every $V_i \in \mathcal{V}$, the pseudonym $\hat{S}_{\pi(i)} = \hat{g}^{s_i}$ is published at position $\pi(i)$ on the public bulletin board. $\hat{g} \in \mathbb{Z}_p^*$ is the so-called *pseudonym generator* and π is an unknown permutation of $\{1, \ldots, m\}$. This step is conducted prior to every voting event.

Definition: Define $g_0 := g$ and $\mathbf{S}_0 = (S_{0,1}, \ldots, S_{0,m}) := (S_1, \ldots, S_m)$. Taking g_0 and $\mathbf{S}_0$ as input, P_1 is responsible for the creation and publishing of g_1 and $\mathbf{S}_1 = (S_{1,1}, \ldots, S_{1,m})$ according to the details given below. If the output of P_1 is verifiably correct, then P_2 uses it for the creation of g_2 and $\mathbf{S}_2 = (S_{2,1}, \ldots, S_{2,m})$, and so on for all pseudonym producers $P_j \in \mathcal{P}$. At the end of the chain, P_n outputs the resulting pseudonym generator $\hat{g} := g_n$ and the permuted list of pseudonyms $\hat{\mathbf{S}} := \mathbf{S}_n = (S_{n,1}, \ldots, S_{n,m})$, which contains V_i's pseudonym $\hat{S}_{\pi(i)} = S_{n,\pi(i)}$ at position $\pi(i)$. The permutation $\pi = \pi_n \circ \cdots \circ \pi_1$ is the result of a sequence of individual permutations π_j, where P_j is responsible for selecting π_j. In the ideal case, in which all n pseudonym producers publish verifiably correct outputs, we obtain thus the following two chains of public values on the public bulletin board:

$$\begin{array}{ccccccccccc} g = g_0 & \rightarrow & g_1 & \rightarrow & g_2 & \rightarrow & \cdots & \rightarrow & g_n = \hat{g}, \\ \mathbf{S} = \mathbf{S}_0 & \rightarrow & \mathbf{S}_1 & \rightarrow & \mathbf{S}_2 & \rightarrow & \cdots & \rightarrow & \mathbf{S}_n = \hat{\mathbf{S}}. \end{array}$$

To produce g_j and $\mathbf{S}_j$ from g_{j-1} and $\mathbf{S}_{j-1}$, respectively, P_j chooses $\alpha_j \in_R \mathbb{Z}_q$ and π_j uniformly at random to compute

$$g_j = g_{j-1}^{\alpha_j},$$
$$S_{j,\pi_j(i)} = S_{j-1,i}^{\alpha_j},$$

for all $i \in \{1, \ldots, m\}$. Obviously, this implies $\hat{g} = g^{\alpha_1 \cdots \alpha_n}$ and thus

$$\hat{S}_{\pi(i)} = S_i^{\alpha_1 \cdots \alpha_n} = (g^{s_i})^{\alpha_1 \cdots \alpha_n} = (g^{\alpha_1 \cdots \alpha_n})^{s_i} = \hat{g}^{s_i},$$

which means that the pseudonyms are evidently generated as intended. Note that V_i can independently compute $\hat{S}_{\pi(i)} = \hat{g}^{s_i}$ using the public pseudonym generator $\hat{g}$ and the secret credential s_i (see Subsection IV-E).

To avoid that the pseudonym producers deviate from the protocol by not choosing the values α_j uniformly at random, we ask them to select α_j and publish $A_j = g^{\alpha_j}$ prior to the pseudonym generation process. Thus, value A_j serves as P_j's commitment to α_j.

Finally, to ensure that the output of each pseudonym producer $P_j \in \mathcal{P}$ is verifiably correct, it must be equipped with a corresponding zero-knowledge proof of correctness Z_j. This proof includes three components, one that proves conformity with the commitment A_j, one that proves the correct computation of g_j, and one that proves correct shuffling. Algorithm 1 shows all the details of what P_j needs to do (assuming that g_{j-1} and $\mathbf{S}_{j-1}$ are correct inputs).

Algorithm 1 Calculate g_j, $\mathbf{S}_j$, Z_j

Require: g_{j-1}, $\mathbf{S}_{j-1}$, α_j, A_j
  $g_j \leftarrow g_{j-1}^{\alpha_j}$
  $\pi_j \leftarrow$ random permutation of $\{1, \ldots, m\}$
  $\mathbf{S}_j \leftarrow$ initialize as m-ary vector
  for all $i = 1, \ldots, m$ **do**
    $S_{j,\pi_j(i)} \leftarrow S_{j-1,i}^{\alpha_j}$
  end for
  $Z_j \leftarrow ZKP\big[(\alpha_j) : (A_j = g^{\alpha_j}) \wedge (g_j = g_{j-1}^{\alpha_j}) \wedge (\bigwedge_{k=1}^{m} \bigvee_{i=1}^{m} S_{j,i} = S_{j-1,k}^{\alpha_j})\big]$
  Post g_j, $\mathbf{S}_j$, Z_j to public bulletin board, keep α_j, π_j secret

For the sake of simplicity, we assumed in Algorithm 1 that all previous pseudonym producers $P_1, \ldots, P_{j-1}$ have correctly fulfilled their tasks and that the input parameters g_{j-1} and $\mathbf{S}_{j-1}$ have thus been computed correctly from g_0 and $\mathbf{S}_0$. By withdrawing this assumption, i.e., by considering the situation where pseudonym producers do miscomputations, choose incorrect inputs, or produce any other type of incorrect outputs, P_j would need to verify all existing proofs Z_1 to Z_{j-1} before executing Algorithm 1. Then, instead of simply taking the outputs of P_{j-1} as input, P_j selects the greatest value $k < j$

such that correct proofs exist for P_k and all its predecessors. Additionally, P_j needs to check that every P_ℓ involved in the chain of correct proofs (i.e., from g_k and $\mathbf{S}_k$ back to g_0 and $\mathbf{S}_0$, respectively) has correctly followed this rule for selecting the input parameters. Note that the same selection rule must be applied at the end of the pseudonymization process for the selection of $\hat{g}$ and $\hat{\mathbf{S}}$ (instead of simply taking g_n and $\mathbf{S}_n$). The length of the corresponding chain of proofs must be greater than or equal to the specified threshold t.

A problem of Algorithm 1 in its simple description is the size of the involved proof, which grows quadratically with the number of voters. As a counter-measure, we may break up the input vector $\mathbf{S}_k$ (and thus $\mathbf{S}_j$) into $\frac{m}{b}$ sub-vectors of size b (suppose m is a multiple of b). Algorithm 1 can then process each of these sub-vectors individually. This reduces the size of the involved proofs and therefore the total running time of Algorithm 1 from $\mathcal{O}(m^2)$ to $\mathcal{O}(m{\cdot}b)$. For a fixed value b, the whole pseudonymization procedure runs then in $\mathcal{O}(m{\cdot}n)$ time.[8]

D. Key Generation for Vote Encryption and Tallying

Objective: For vote encryption and tallying, corresponding keys of a secure (t, n)-threshold ElGamal cryptosystem are generated. The private key d is shared among the members of $\mathcal{T}$ and the corresponding public key e is published. This step is conducted prior to every voting event.

Definition: An appropriate protocol for secure distributed key generation based on *Shamir's Secret Sharing Scheme* [25] is proposed in [18]. To apply it in the context of our e-voting protocol, we need a second generator $h \in_R G_q \setminus \{1\}$ of the same subgroup G_q, which is jointly selected at random by the members of $\mathcal{T}$.[9] At the end of the protocol, a public key $e = h^d \in G_q$ is published. The corresponding private key $d \in_R \mathbb{Z}_q$ is shared among the members of $\mathcal{T}$ and can only be computed by a coalition of size t or greater. Any smaller coalition has no advantage over a single adversary who tries to compute d from h and e without owning a share. Parties that deviate from the key generation protocol will be detected and disqualified by the others.

E. Vote Casting

Objective: An ElGamal encryption $w_i = (x_i, y_i)$ of vote v_i is cast to the public bulletin board along with a proof to show that its owner is an eligible voter who owns a receipt.

Definition: Voter V_i calculates the pseudonym $\hat{S}_{\pi(i)} = \hat{g}^{s_i}$ and the ElGamal encryption $w_i = (x_i, y_i) = (h^{s_i}, v_i{\cdot}e^{s_i})$ of the vote v_i. It is crucial that s_i is used as the ElGamal randomness (see Subsection IV-F). Using the pseudonym generator $\hat{g}$, the voter further computes a zero-knowledge proof $z_i = ZKP\big[(s_i) : (\hat{S}_{\pi(i)} = \hat{g}^{s_i}) \wedge (x_i = h^{s_i})\big]$ and then posts $(\hat{S}_{\pi(i)}, w_i, z_i)$ to the public bulletin board through an anonymous channel. The vote cast is accepted and published on the board, if $\hat{S}_{\pi(i)}$ is a valid pseudonym enlisted in $\hat{\mathbf{S}}$ and if the verification of z_i yields *true*.[10] If several votes are cast under the same pseudonym, only one of them is kept according to some policy. Note that the procedure so far guarantees that the encrypted vote w_i is verifiably owned by an eligible voter, since only members of the voter roll are assigned a pseudonym. Subsection IV-F shows how V_i can use s_i to prove ownership of the vote and that s_i is actually a receipt.

F. Proofs of Eligibility and Ownership

Objective: V_i is able to either prove not having cast a vote or to identify the encrypted vote $w_i = (x_i, y_i)$ on the public bulletin board. If necessary, V_i may even be able to verifiably disclose v_i (e.g. with regard to the possible application of the second revocation procedure described in [4]).

Definition: The voting officials identify the public credential S_i on the public bulletin board as V_i authenticates at the polling station. V_i then reveals the pseudonym $\hat{S}_{\pi(i)}$ together with $ZKP\big[(s_i) : (S_i = g^{s_i}) \wedge (\hat{S}_{\pi(i)} = (\hat{g}^{s_i})\big]$ as a proof of correctness. If on the public bulletin board there is no vote associated with $\hat{S}_{\pi(i)}$, V_i has proven the eligibility to cast a paper vote. Otherwise, V_i is clearly the owner of the encrypted vote associated with $\hat{S}_{\pi(i)}$.

If the applied revocation procedure requires v_i to be revealed, V_i can use the secret credential s_i as a receipt to present $ZKP\big[(s_i) : (x_i = h^{s_i}) \wedge (\frac{y_i}{v_i} = e^{s_i})\big]$. This proves that v_i has been revealed truthfully. By previously handing out s_i to a coercer, V_i might not know v_i, but it can easily be calculated as $\frac{y_i}{e^{s_i}}$. Due to the zero-knowledge property of Σ-protocols, the credential s_i can be reused for subsequent voting events.

G. Tallying

Objective: The result of the tally is published and provably correct.

Definition: At least t members of $\mathcal{T}$ publicly reveal their share of d and prove having done so correctly according to [18]. Now anybody could efficiently calculate d using any set of at least t shares, and decrypt all votes cast to compute the final outcome of the voting event.[11]

V. Discussion

In its hybrid context, we relate the protocol as defined in the previous section to the requirements on e-voting systems as presented in Subsection II-A.

[8] To guarantee a sufficiently large space of possible permutations over $\{1, \ldots, m\}$ even for a small value b (e.g. $b = 2$), we may apply some prescribed pattern to assign different sets of sub-vectors to each pseudonym producer. In this way, up to $(b!)^{\frac{m \cdot n}{b}}$ different random permutations are possible and equally likely. For example, $b = 2$ yields $(\sqrt{2})^{m \cdot n}$ possible permutations. This number is far less than $m!$, the number of all permutations over $\{1, \ldots, m\}$, but still large enough for a sufficiently randomized shuffling (exponential in both m and n).

[9] Particularly, $\log_g(h)$ needs to be unknown, because otherwise adversaries could link votes to their owners (see vote casting step as given in Subsection IV-E). As described in [18], a generic distributed coin flipping protocol can be applied to serve that purpose.

[10] The board may also publish all incoming posts and delegate the verification and filtering to the talliers.

[11] In practice, the authorities could be responsible for the decryption, and everybody could verify that they have done so correctly.

A. Accuracy

Given that votes reach the public bulletin board in an unchanged state, the *integrity*, *completeness* and *soundness* requirements are trivially met by an appropriate definition of the public bulletin board and the fact that votes can be decrypted by any observing party at tallying-time. However, voters are required to verify that their votes actually reach the public bulletin board in an unchanged state and to react accordingly otherwise, i.e. by resending or even revoking their vote.[12]

B. Democracy

Eligible voters are assigned a public credential and a pseudonym. By the definition of the pseudonym generation process, anybody can verify that each eligible voter is assigned his designated unique pseudonym correctly. Assigning pseudonyms to citizens not enlisted in the voter roll is clearly impossible. Thus the requirement *eligibility* is met.

Given that votes need to be publicly associated with a pseudonym, multiple voting is excluded, thus *uniqueness* is achieved.

C. Privacy

The requirement *anonymity* is achieved if a vote cannot be linked back to its owner. By sending their vote through the anonymous channel and relating it to their pseudonym, voters anonymously authenticate as eligible voters without disclosing their identity. Thus, the votes are detached from their senders.

For *anonymity* to hold, we additionally need to rely on the assumption that pseudonyms cannot be efficiently linked back to their corresponding public credential. Given that at least two members from the group of pseudonym producers $\mathcal{P}$ withhold their secret value α_j used at pseudonym generation, this assumption is strongly related with the *Decision Diffie-Hellman Assumption* DDH.[13] Given any two vectors $\mathbf{S}_k, \mathbf{S}_\ell$ and respective values g_k, g_ℓ, where $k < \ell$ and $\alpha_{k+1} \cdots \alpha_\ell = \log_{g_k}(g_\ell)$ is unknown, DDH inherently yields that it is computationally hard to identify $S_{k,w}$ for given $S_{\ell,v}$, such that $S_{\ell,v} = S_{k,w}^{\alpha_{k+1} \cdots \alpha_\ell}$ holds. Particularly, given the full list of pseudonyms, it is infeasible to identify the pseudonym $\hat{S}_{\pi(i)}$ that corresponds with a given public credential S_i and vice-versa. Further, it seems that an attacker strategy to apply appropriate transformations to votes cast and any combination of lists $\mathbf{S}_0, \ldots, \mathbf{S}_n$ as well as respective $g_0, \ldots, g_n$ to reproduce the applied permutations, would yet require DDH to break.

As mentioned above, we do not address the possibility of phishing attacks on a protocol level. We find this justified by the fact that the issue is not specific to e-voting and that we have no knowledge of an existing e-voting protocol that provides privacy under the presence of malware. However, we believe that the problem can be solved to a satisfying degree on an implementation level of the protocol, namely by using trusted offline devices, possibly based on smart-card technology, for critical computations. We consider the fact that the protocol allows to be implemented in a secure way to yield *anonymity* as a quality attribute.

In pure e-voting systems, the requirement *receipt-freeness* is stated as a precondition to coercion-resistance. In Subsection II-C we argue why hybrid voting systems offer coercion-resistance by definition. Further, we argue why e-voting systems that guarantee a receipt to vote owners are candidates for being used as the electronic sub-component of a hybrid context. Therefore, the protocol is actually designed to offer receipts to vote-owners.

D. Verifiability

Using the secret credential s_i and the public values of the ElGamal PKI, voter V_i can re-calculate the encryption of v_i and verify that it is shown on the the public bulletin board.

For tallying, at least t talliers of $\mathcal{T}$ reveal their share of their group's private key d, and prove having done so correctly. The talliers then publish d, use it to decrypt each vote, and publish the plaintext votes associated with their encryption and the pseudonym of their owner. Thus, voter V_i can verify that the vote is decrypted correctly. Since it can be verified that the talliers have revealed their shares truthfully, V_i can calculate d and use it to decrypt all votes. By comparing the tally of the decrypted votes with the published tally, V_i knows that all votes (including v_i) have been counted correctly. Thus, the requirements *individual verifiability* and *universal verifiability* are met.

Relating the plaintext votes to their encryption and their owner's pseudonym, even after tallying, creates the highest possible sense of trust among voters regarding the accuracy of the tally. This is particularly relevant considering the majority of voters who do not have the background it takes to understand a potentially complex tallying procedure that detaches plaintext votes from their encryptions.

E. Fairness

Fairness is achieved by leaving the plaintexts of the encrypted votes unrevealed until the polls are closed. Recall that it takes at least t members of $\mathcal{T}$ to decrypt votes. By assuming

[12] Vote revocation without coercion would be necessary, for instance when assuming the presence of malware running on voters' platforms that may exploit the malleability property of the ElGamal cryptosystem to modify the vote. The presence of malware may also affect the protocol's performance with regard to the *privacy* and *fairness* requirements. Although we could suggest some modifications to the protocol to tackle some of the security concerns, it seems that privacy cannot be guaranteed when assuming untrusted platforms. However, this problem applies to all internet applications and is not specific to e-voting. Accordingly, we assume that voters own a trusted device for doing sensitive computations. This could indeed be realized by exporting the computations related to vote encryption to a smartcard and sending a non-malleable encryption of the encrypted vote and the proof to the electronic ballot-box. Thus, phishing attacks are avoided and votes would reach the public bulletin board in an unchanged state. Nevertheless, the voter would need to check that the vote cast reaches the public bulletin board when assuming unreliable anonymous channels.

[13] The decisional Diffie-Hellman assumption (DDH) states that it is computationally hard to distinguish the triples (g^a, g^b, g^{ab}) and (g^a, g^b, g^c), where a, b, c are chosen at random from $\mathbb{Z}_q$ and G_q is a finite cyclic group of order q generated by g. Although this is not generally true, DDH is believed to hold if G_q is chosen as the q-order subgroup of $\mathbb{Z}_p^*$ generated by g, such that p and $q = (p-1)/k$ are large primes. This restriction corresponds with the suggested settings in this paper.

the number of untrusted members smaller than t, as stated in Subsection IV-A, the event of premature decryption of votes is ruled out, and thus the *fairness* requirement is met.

F. Coercion-Resistance

As shown in Subsection IV-F, the protocol meets the requirements imposed on the electronic sub-component of a hybrid voting system. The definition of a hybrid voting system directly yields *coercion-resistance*, as explained in Subsection II-C.

VI. Conclusion

Governments will not immediately replace their traditional paper-based voting scheme with a pure internet e-voting system. Instead, both voting channels will need to be integrated. We exploit this natural setting to propose hybrid systems to overcome the danger of vote buying and voter coercion to which integrated voting systems are subjected. Thus, voters can trust the published outcome of votes as reflecting the actual, unbiased political preference of the voting community as a whole.

We have shown that the presented protocol satisfies the requirements on the electronic subsystem of a hybrid voting system. Moreover, it provides the key requirements expected from a voting system: Accuracy, democracy, privacy, verifiability and fairness are achieved even if the protocol is applied in a stand-alone voting system. It also provides coercion-resistance, when operated in its hybrid context.

The protocol will most likely create a high degree of trust regarding the correctness of the published vote outcome. Voters do not need to be proficient in any cryptographic basics. Just by seeing their pseudonym associated with their decrypted vote on the public bulletin board, they will know that their vote has been counted correctly.

References

[1] Die Bundesbehörden der Schweizerischen Eidgenossenschaft, "Bericht über den vote électronique: Chancen, risiken und machbarkeit elektronischer ausübung politischer rechte," *Bundesblatt*, vol. 154, no. 5, pp. 645–700, 2002.

[2] L. Loeber, "E-voting in the Netherlands: from general acceptance to general doubt in two years," in *3nd International Workshop on Electronic Voting*, ser. Lecture Notes in Informatics, R. Krimmer and R. Grimm, Eds. Bregenz, Austria: Gesellschaft für Informatik E.V., 2008, pp. 21–30.

[3] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in *WPES'05, 4th ACM Workshop on Privacy in the Electronic Society*, V. Atluri, S. De Capitani di Vimercati, and R. Dingledine, Eds., Alexandria, USA, 2005, pp. 61–70.

[4] O. Spycher, R. Haenni, and E. Dubuis, "Coercion-resistant hybrid voting systems," in *4th International Workshop on Electronic Voting*, R. Krimmer and R. Grimm, Eds., Bregenz, Austria, 2010.

[5] L. F. Cranor and R. K. Cytron, "Design and implementation of a practical security-conscious electronic polling system," Washington University, Tech. Rep. WUCS-96-02, 1996.

[6] R. Haenni, E. Dubuis, and U. Ultes-Nitsche, "Research on e-voting technologies," Bern University of Applied Sciences, Tech. Rep. 5, 2008 .

[7] J. Benaloh and D. Tuinstra, "Receipt-free secret-ballot elections," in *STOC'94, 26th Annual ACM Symposium on Theory of Computing*, Montréal, Canada, 1994, pp. 544–553.

[8] J. Skripsky, "Minimal models for receipt-free voting," Semester Project, ETH Zürich, 2002.

[9] S. Delaune, S. Kremer, and M. Ryan, "Coercion-resistance and receipt-freeness in electronic voting," in *CSFW'06: 19th IEEE workshop on Computer Security Foundations*, Venice, Italy, 2006, pp. 28–42.

[10] H. L. Jonker and E. P. Vink, "Formalizing receipt-freeness," in *ISC'06, 9th Information Security Conference*, ser. LNCS 4176, Samos, Greece, 2006, pp. 476–488.

[11] T. Okamoto, "Receipt-free electronic voting schemes for large scale elections," in *5th International Security Protocols Workshop*, ser. LNCS 1361, B. Christianson, B. Crispo, T. M. A. Lomas, and M. Roe, Eds., Paris, France, 1997, pp. 25–35.

[12] M. Hirt and K. Sako, "Efficient receipt-free voting based on homomorphic encryption," in *EUROCRYPT'00, International Conference on the Theory and Applications of Cryptographic Techniques*, ser. LNCS 1807, G. Goos, J. Hartmanis, and J. van Leeuwen, Eds., Bruges, Belgium, 2000, pp. 539–556.

[13] E. Magkos, M. Burmester, and V. Chrissikopoulos, "Receipt-freeness in large-scale elections without untappable channels," in *I3E'01, 1st IFIP Conference on towards the E-Society*, B. Schmid, K. Stanoevska-Slabeva, and V. Tschammer, Eds., vol. 202, 2001, pp. 683–694.

[14] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo, "Providing receipt-freeness in mixnet-based voting protocols," in *ICISC'03, 6th International Conference on Information Security and Cryptology*, ser. LNCS 2971, G. Goos, J. Hartmanis, and J. van Leeuwen, Eds., Seoul, South Korea, 2003, pp. 245–258.

[15] Z. Xia and S. Schneider, "A new receipt-free e-voting scheme based on blind signature," in *WOTE'06, IAVoSS Workshop on Trustworthy Elections*, Cambridge, U.K., 2006, pp. 127–135.

[16] T. Moran and M. Naor, "Receipt-free universally-verifiable voting with everlasting privacy," in *CRYPTO'06, 26th Annual International Cryptology Conference on Advances in Cryptology*, ser. LNCS 4117, C. Dwork, Ed., Santa Barbara, USA, 2006, pp. 373–392.

[17] S. S. M. Chow, J. K. Liu, and D. S. Wong, "Robust receipt-free election system with ballot secrecy and verifiability," in *NDSS'08, 15th Network and Distributed System Security Symposium*, San Diego, USA, 2008, pp. 81–94.

[18] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," in *EUROCRYPT'99, International Conference on the Theory and Application of Cryptographic Techniques*, ser. LNCS 1592, J. Stern, Ed., Prague, Czech Republic, 1999, pp. 295–310.

[19] E. Bangerter, "Efficient zero-knowledge proofs of knowledge for homomorphisms," Ph.D. dissertation, Fakultät für Elektrotechnik und Informationstechnik, Ruhr-Universität Bochum, Germany, 2005.

[20] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[21] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *CRYPTO'92, 12th Annual International Cryptology Conference on Advances in Cryptology*, ser. LNCS 740, E. F. Brickell, Ed., Santa Barbara, USA, 1992, pp. 89–105.

[22] D. Chaum, "Untraceable electronic mail, return addresses and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–88, 1981.

[23] S. Schechter, T. Parnell, and A. Hartemink, "Anonymous authentication of membership in dynamic groups," in *FC'99, 3rd International Conference on Financial Cryptography*, ser. LNCS 1648, M. K. Franklin, Ed., Anguilla, British West Indies, 1999, pp. 184–195.

[24] K. Sako, S. Yonezawa, and I. Teranishi, "Anonymous authentication: For privacy and security," *NEC Journal of Advanced Technology*, vol. 2, no. 1, pp. 79–83, 2005.

[25] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.